

Sevgili okuyucular,

Bu sayımızda sizlere **Uzman Sistemler** olarak anılan ve belirli bir alanın önde gelen uzmanlarından derlenen bilgilere dayanarak düzenlenen çok sayıdaki "eğer...ise...dir" biçimindeki kurallardan oluşan bilgi tabanlı yazılım sistemleri hakkında bilgi vereceğiz. Bu konudaki yazının hazırlanmasında emeği geçen Ortadoğu Teknik Üniversitesi Elektrik Mühendisliği Bölümü öğrencilerinden Hakan Karakaş'a teşekkür ederiz.

Ayrıca, Selçuk Üniversitesi Elektrik Mühendisliği Bölümü öğrencilerinden 0422-72-70 no'lu üyemiz Murat Saygı'nın hazırladığı ve elinizdeki oyunlarla ilgili bir veri tabanı oluşturmakta kullanabileceğiniz bir program veriyoruz.

Sizlerin de kısa program, yazı duyuru veya bilgisayar konusu ile ilgili grafik, karikatür, fıkra ya da benzeri şeylerle klübe katkılarınızı bekliyoruz. Klübümüze üye olmak isteyenler için yazışma adresimiz:

Emrehan Halıcı,
Bilim ve Teknik Dergisi,
Bilgisayar Klübü,
Atatürk Bulvarı, No: 221
Kavaklıdere, Ankara

UZMAN SİSTEMLER

Uzman sistemler (expert systems) mühendislik uygulamaları, bilimsel çözümleme, tıbbi tanımlama, mali çözümleme gibi normalde bu konuda uzmanlaşmış kişiler tarafından çözümlenen sorunların bilgisayar yardımı ile çözümlenmesinde kullanılan sistemlerdir. Uzman sistemler, hangi uzmanlık konusundaki problemlerin çözümünde kullanılacaklarsa, o konuda oluşturulmuş özel bilgi tabanlarına (knowledge base) ihtiyaç duyarlar. Bu bilgi tabanının yanında sahip oldukları bu bilgileri yorumlayacak ve verilen probleme uygulayacak mantık yürütme mekanizmalarından en az birini içermeleri de zorunludur. Bu açıdan bakıldığında uzman sistemler geniş bir çerçevede yapay zekâ uygulamalarının temsil ederler.

Uzman sistemler karmaşık yapay zekâ programlarıdır. Programın kullanıldığı kaynak bilgiler ise birtakım kural setleri halinde saklanır ve kural setleri genellikle onları yorumlayacak olan bir mantık yürütme sistemi ile eşlenir. Sonuçta program, sahip olduğu bilgi tabanına ve kullandığı mantık yürütme sistemine göre verilen problem için bir çözüm önerir. Uzman

sistemler karar verirken kullandıkları bilgi tabanlarına bağlıdır. Bu yüzden verilen bir problem için bir sonuç bulamamaların yanında yanlış sonuçlara da ulaşabilirler. Dolayısıyla ile, uzman sistemlerin verilen problem hakkında her zaman doğru karar vermesini beklemek pek doğru olmaz.

Örnek olarak tıbbî tanı yapacak olan bir uzman sistemin bilgi tabanının nasıl tanımlanacağını görelim. Böyle bir uzman sistem hastalık belirtilerinden oluşan bir bilgi tabanına ihtiyaç duyar ve gerekli bilgilere aşağıdaki gibi tanımlanabilir;

eğer;

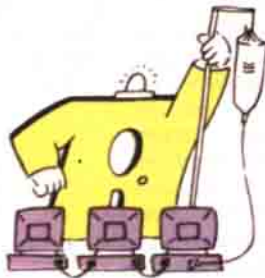
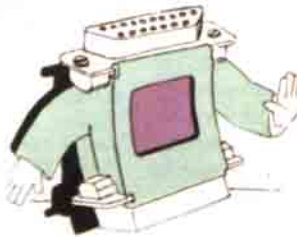
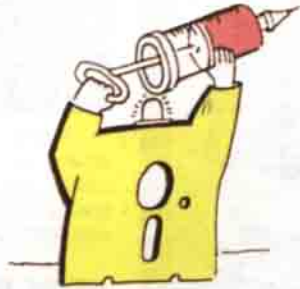
- hastanın ateşi var
- hastanın midesi bulantıyor ise; tanı = = => % 70 güneş çarpması
- hastanın ten rengi soluk ise; tanı = = => % 30 zehirlenme

Buna benzer binlerce tanı tümesinden oluşan böyle bir bilgi tabanına, hastanın ateşli olduğu ve midesi bulantısını belirtmek üzere

hasta (ateş, mide bulantısı)?

türünden bir soru sorulursa program bilgi tabanı ve mantık yürütme mekanizmalarını kullanarak sonuçta tanı olarak 'güneş çarpması' nı önerebilir. Doğal olarak böyle kısıtlı bir sorgulama ile benzer özellikleri taşıyan yüzlerce hastalık bulunabilir. Bu ve bunun gibi karışıklıkları önlemek için uzman sistemler karmaşık mantık algoritmaları ve çeşitli bilgi tanımlama metodları kullanırlar.

Genellikle LISP ile yazılan uzman sistemler, iki ana kısımdan oluşurlar: Tanıtıcı kurallar seti ile yorumlayıcılar. Yazılan birçok sistemden sonra hepsinde ortak olan birçok kısmın bulunduğu fark eden yapay zekâcılar birçok sistemde aynı olan yorumlayıcı kısımlarını programdan ayırarak ayrı bir ad altında topladılar. Kabuk (shell) adı verilen bu programlar sayesinde yeni sistemler oluşturmak daha kolaylaştı. Çünkü artık yeni bir sistem oluşturmak için sadece yeni problemi tanımlayan bilgilerin önceki bilgi tabanına



eklenmesi yeterli olmaya başladı. Böylece uzman sistem oluşturan uzmanlara büyük bir rahatlık ve esneklik sağlanmış oldu. Şu anda ticarî olarak kullanılan birçok uzman sistem kabuğu, oluşturulmakta olan diğer birçok uzman sistemin tabanını oluşturmaktadır.



Tıbbî tanı yapacak olan bir uzman sistem, hastalık belirtilerinden uzman bir bilgi tabanına ihtiyaç duyar.

Bir uzman sistem nasıl oluşturulur? Genellikle bir bilgi mühendisi, uzman sistemin bilgi tabanını oluşturacak bilgileri elde etmek üzere alandaki bir uzmanla görüşür ve daha sonra bu bilgileri uzman sistemi oluşturacak kurallara çevirir. Bu şekilde kurulan sistem yinelenmeli denemelerle uzman seviyeyi yakalayana kadar denenin, eksikleri tamamlanır. Bu tür bir yol çok zaman alıcı ve pahalıdır. Ne yazık ki, henüz tamamen otomatik bilgi toplama sistemlerine (knowledge acquisition) sahip değiliz; ancak yine de, uzman kişi ile etkileşerek ondan uzman sistemler için gerekli bilgileri en verimli şekilde alan birçok programa sahibiz. Bu programlar, gerekli bilgilerin kolayca bilgi tabanına girilmesine, girilen bilgilerin tutarlılığının sağlanmasına ve bilgilerin bütünlüğünün garantielenmesine yardımcı olurlar.

Özet olarak uzman sistemler hakkında şunları söyleyebiliriz:

* Uzman sistemler, başarılarını sahip oldukları geniş özel kaynaklı bilgi tabanlarından alırlar.

* Başarılı bir uzman sistem, özel bir alanda iyi tanımlanmış bir bilgi tabanı gerektirir. Genel ve ta-

nımlanması zor bilgi tabanlarına dayanan uzman sistemler hataya daha açıktırlar.

* Uzman sistemlerin oluşturulması bu alanda uzman kişilerin bilgilerini sisteme aktarmak için çalışmalarını gerektirir.

* Gerekli olan bilgi miktarı yapılan iş ile ilgilidir. Sistem birkaç set kuraldan oluşacağı gibi milyonlarca kural içeren bilgi tabanları oluşturulabilir.

* Kullanılacak denetim mekanizması, uygulamadan uygulamaya değişebilmektedir; ancak yine de oluşturulan bir sistem diğer farklı sistemlerin oluşturulmasında taban olarak kullanılabilir.

Sonuç olarak, uzman sistemler bize birçok kolaylık sağlamlarına rağmen daha çözülecek pek çok sorunu da beraberinde getirmektedirler. Yapılan birçok araştırma ile bu sorunlar giderilmeye çalışılıyor; ancak uzman sistemlerin gerçek uzmanların yerlerini alması için üstesinden gelinmesi gereken daha birçok sorun var. Bunlardan en önemlisi ise, sistemlere sağduyulu davranmayı öğretmek gözüküyor.

BİLGİSAYARLAR VE 4. BOYUT

Aldığımız 3-boyutlu uzay yerine, yaşadığımız evren başka boyutların eklenmesiyle 4, 5 6. boyutlu hale çıkarılabileceydi, acaba nasıl gözükürdü. Ne yazık ki 3-boyuttan sonrasını algılayabilmek bizler için oldukça güç.

Yıllardır matematikçiler 4-boyutlu uzayda yer alan cisimlerin iki ya da üç boyutlu uzaydaki izdüşümlerinin nasıl olduğunu inceleyerek bu cisimlerin çeşitli özelliklerini anlamaya çalışıyorlar. Bu işlerin büyük bir kısmını 3-boyutlu evrenin 2-boyutlu bir fotoğrafa izdüşümünün alındıktan sonra fotoğraftan 3-boyutlu evrenin nasıl olduğunu anlamaya çalışmaya benzetebiliriz.

1960'lı yıllarda, 4-boyutlu uzayın özelliklerini bilgisayar grafiği kullanıla-

rak göstermek üzere A. Michael Noll tarafından Bell laboratuvarlarında bir proje başlatılmıştı. Ancak bu çalışmaların 25 yıl daha öncesinde Brown üniversitesinden Thomas Banchoff adlı bir matematikçi, yine aynı üniversiteden Charles Strauss adlı bir bilgisayarının hazırladığı programı kullanarak 4-boyutlu uzayda dönmekte olan cisimlerin izdüşümlerini bilgisayar ekranında göstermekte kullanmıştı. Sınırlı hesaplama güçleri dolayısıyla bu ilk bilgisayarlar, 4-boyutlu cisimlerden sadece çok basit olanların izdüşümlerini görüntüleyebiliyorlar ve etkileşimli görüntüleme yapılmaya çalışıldığı 3-boyutlu cisimlerin bile gösteriminde yetersiz kalıyorlardı.

Günümüzde, çok güçlü donanımlarla, 3-boyutlu cisimleri ekranda göstermek üzere geliştirilen yeni yazılım tekniklerinin bir araya gelmesi, araştırmacıların bu konudaki çok ilginç görüntüleri gözlemelerine imkân sağlıyor. Banchoff, şimdi kullandığı güçlü sistemde, 4-boyutlu cisimlerin değişik açılardan kesitlerini alarak, sürekililiğin kesintiye uğradığı tekil noktalarını (singularity) bulmaya çalışıyor. İki boyutlu uzayda bakacak olursak köşeler, kenarlar bu tür tekil noktalara karşılık geliyorlar. Banchoff'un çalışmaları temelinde, 4-boyutlu uzaydaki yüzeylerin izdüşümlerindeki tekilliklerin cisimler döndürüldükçe nasıl etkilendiğini gözlemek, bu tekillikleri cisimler döndürüldükçe nasıl etkilendiğini gözlemek, bu tekillikleri karşılaştırmak ve sınıflandırmaktan oluşuyor.



Bilgisayar grafiği konusundaki ilerlemeler, bu Klein şişesi resiminde açıkça hissediliyor. Beyaz çizim 20 yıl önce, renkli olansa bir kaç ay önce hazırlanmış.

Resimde, Banchoff ve öğrencilerinin ürettikleri ilginç bir Klein Şişesi yeralıyor. Biri Klein Şişesi çesnek bir silindirin bir kez bükülerek iki ucunun birleştirilmesinden elde ediliyor. Ancak, birleştirme işlemi simit benzeri bir halka getirmek üzere yapılmıyor, bunun yerine uçlardan birisi döndü-

rülerek kendi içerisinde geçilecek diğerine yaklaştırılıyor. Banchoff Klein şişesinin bölümlerinin 4-boyutlu uzayda diğer bölümlerle olan ilişkilerini göstermek üzere renk kodlaması kullanmış. Bu kodlamaya göre kırmızı en yüksek ilişkiyi, yeşil ise en düşük ilişkiyi gösteriyor.

BİLGİSAYAR DÜNYASINDAN HABERLER

ISCIS VIII, 3-5 KASIM 1993'TE ANTALYA'DA YAPILACAK

Türkiye'de bilgisayar konusunda düzenlenen en önemli uluslararası konferans olan ISCIS, International Symposium on Computer and Information Sciences, bu yıl Antalya'da 3-5 Kasım 1993 tarihleri arasında yapılacak. Bu konferans, Türkiye'nin çeşitli üniversitelerinde bilgisayar konusunda çalışan kişileri bir araya getirmenin yanı sıra yurt dışından gelen birçok bilim adamıyla da bilgi alışverişinde bulunma imkânı sağlıyor. 8 yıldan beri düzenli olarak sonbaharda yapılan konferansın organizasyonunu bu yıl Fransa'dan Üniversite Rene Descartes'e bağlı olan Ecole des Hautes Etudes en Informatique (EHEI), Türkiye'den Orta Doğu Teknik Üniversitesi (ODTÜ) ve Amerika'dan Duke Üniversitesi üstleniyor. IEEE Computer Society Türkiye Chapter (IEEE CS TC) işbirliğinde bulunduğu sempozyumda, "Algoritmalar ve Teori", "Bilgisayar Mimarileri", "Dağıtık ve Paralel Sistemler", "Yapay Zekâ", "Gerçek-Zaman Sistemleri", "Yapay Sinir Ağları", "Veri Tabanları", "Yazılım Mühendisliği", "Multimedya ile İletişim", "Örüntü Tanıma ve Görüntü İşleme" "Bilgisayar Haberleşme Ağları" konularında makalelerin kabul edileceği konferansa makale gönderme tarihi 30 Haziran 1993'te sona eriyor. Konferans ile ilgilenenler, ISCIS VIII, Uğur Halıcı, Elektrik ve Elektronik Müh. Böl., ODTÜ, 06531, Ankara,

Tel: (4) 210 10 00/2301/2333

Fax: (4) 210 12 61

E-posta: iscis@trnetu.bitnet adresinden daha ayrıntılı bilgi isteyebilirler.

2. TÜRK YAPAY ZEKÂ VE YAPAY SİNİR AĞLARI SEMPOZYUMU

Sizlere daha önce duyurduğumuz Yapay Zekâ ve Yapay Sinir Ağları Sempozyumu'nun ikincisi bu ay, 24-25 Haziran 1993 tarihlerinde İstanbul'da Boğaziçi Üniversitesi'nde yapılacak. Boğaziçi Üniversitesi, Bilkent Üniversitesi, IEEE Computer Society Türkiye Kolu, Orta Doğu Teknik Üniversitesi ve TÜBİTAK işbirliği ile desteklenen sempozyumdaki bildiri konuları arasında Benzetim Ortamları, Bilgi Gösterimi, Denetim ve Planlama, Doğal Dil İşleme, Donanım Gerçeklemeleri, Endüstriyel Uygulamalar, Görme ve Görüntü İşleme, İşaret İşleme, Konuşma İşleme, Kuramsal Temeller, Öğrenme, Örüntü Tanıma, Programlama Dilleri, Robotik, Sağduyusal Uslamlama, Sosyal Yasal ve Aklaklı Yönler, Teorem İspatlama, Uzman Sistemler, Zeki Veritabanları yer alıyor. Sempozyum ile ilgili daha ayrıntılı bilgi almak isteyenler,

Yrd.Doç.Dr. Levent Akın, Bilgisayar Mühendisliği Bölümü, Boğaziçi Üniversitesi, 80815, Bebek, İstanbul

Tel: (1) 263 15 00/1323,

Faks: (1) 265 84 88,

E-posta: yz@trboun.bitnet adresine başvurabilirler.

ÜYELERDEN

071-71-41 Mehmet Akgün
Dumlupınar Mah. 608. Sok.
No. 14, 41660, Gölcük, İzmit

Boğaziçi Üniversitesi Elektronik Mühendisliği Bölümü'nde okuyan üyemiz Basic ve Pascal biliyor. Elektronik, bilgisayar, astronomi ile ilgileniyor. Elektronik baskı devre programları kullanıyor.

072-75-58 Öner Büyükyıldız
Atatürk Cad. Cumhuriyet Mah.
No. 89-A 58500 Yıldızeli, Sivas

Sivas Teknik ve Endüstri Meslek Lisesi Bilgisayar Bölümü'nde okuyan üyemiz Basic biliyor, WordStar, Smart, Microsoft Windows kullanıyor.

079-76-03
Ahmet Büyükerkmen
Ordu Bulvarı TMO Arkası,
Doktorlar Sitesi, A Blok D.7,
03200 Afyon

Kocatepe Anadolu Lisesi'nde okuyan üyemiz, 5 yıl önce bilgi-

BİLGİSAYAR KLÜBÜ ÜYELERİ

Üye numaraları sıra no-doğum tarihi-il biçimidir. İki adet resimler eksik olanlar (r) ile gösterilmiştir. Bu üyelerimizden, en kısa zamanda arkasına isimlerini yazdıkları resimlerini bekliyoruz.

- 0181-70-34 Nebahat Yıldız
- 0182-73-52 Nebi Yüksel
- 0183-63-01 Mehmet Nuri Çiftarslan
- 0184-79-67 Gökhan Toptan
- 0185-63-34 Levent Aslan
- 0186-67-37 Haldun Sekenderiz
- 0187-74-06 İnanç Koyuncu
- 0188-73-34 Sinan Dora
- 0189-74-03 Ercüment Bozdemir
- 0190-75-41 Celal Mızrak
- 0191-75-34 Emrah Alpa
- 0192-71-01 Mahmut Ünal
- 0193-74-34 Kadriye Pamuk (r)
- 0194-71-32 Gültekin Uzunurgancı
- 0195-77-35 Yasemin Yıldız
- 0196-74-37 Kemal Sakkaoğlu (r)
- 0197-76-51 Alparslan Deniz (r)
- 0198-76-66 Kudret Şimşek (r)
- 0199-75-66 Bektaş Eraslan (r)
- 0200-77-34 Melih Gökhan Demir (r)

PROGRAM

Selçuk Üniversitesi, Elektrik Mühendisliği Bölümü öğrencilerinden 0422-72-70 no'lu üyemiz Murat Saygı'nın hazırladığı QBASIC Oyun Kayıt Programı'ndan kısaltarak düzenlediğimiz bölümleri bulacaksınız. Bu programı elinizdeki oyunlarla ilgili bilgileri saklamakta kullanabilirsiniz. Bu program üzerinde ufak değişiklikler yaparak, adres veya telefon listesi tutmak gibi amaçlarla da kullanabilirsiniz. Program sırasında girilen oyun bilgileri OYUN.DAT isimli bir veri dosyasına saklanmaktadır. Dolayısıyla bu bilgiler, Oyun programından çıkılsa veya bilgisayar kapatılsa bile kalıcı olmaktadır, programa daha sonra girildiğinde daha önce girilen bilgilere ulaşılabilir. Oyun bilgileri üzerinde yapılabilecek işlemler bir menü halinde sunulmaktadır. Menüde, Oyun Kayıt, Oyun Arama, Oyun Silme, Oyun Değiştirme, Programdan Çıkış işlemlerinden oluşmaktadır. Dergimize bize ayrılan yerin kısıtlı olması dolayısıyla, bunlardan Silme ve Değiştirme işlemlerini gelecek sayımızda yayınlayacağız. Programınız bu halyle yine çalışıyor durumda olacak, ancak 3. ve 4. seçeneklerde hiçbir şey yapmayacaktır.

sayar programcılığına ilgi duymaya başlamış. İllerinde düzenlenen bir bilgisayar kursuna katılarak Basic, Cobol, Dbase, Qbasic, Fortran dillerini öğrenmiş.

082-56-67

Hüseyin Yüksekay
Dağüstü Köyü,
Çaycuma, Zonguldak

Çaycuma'nın Dağüstü köyünde öğretmenlik yapan üyemiz dama ve satranç oynamayı seviyor. ZX Spectrum Sinclair bilgisayarı var. Bilgisayarda satranç programlarına ilgi duyuyor.

```
5 ON ERROR GOTO 5000
10 CLS : KEY OFF: FOR I = 1 TO 10: KEY I,": NEXT
30 LOCATE 10, 30: PRINT "ŞİFREYİ YAZINIZ.": LOCATE 10, 52: INPUT "": Q$
40 IF Q$ = "12345" THEN CLS : LOCATE 10, 30: PRINT "BEKLEYİNİZ." ELSE GOTO 30
50 OPEN "R", #1, "OYUN.DAT", 64
60 FIELD #1, 10 AS OAS, 20 AS AKŞ, 8 AS AS, 8 AS CS, 1 AS OZS
70 CLS : KEY OFF:
80 LOCATE 2, 25: PRINT "
90 LOCATE 3, 25: PRINT " OYUN KAYIT İŞLEMİ "
100 LOCATE 4, 25: PRINT "
110 LOCATE 5, 25: PRINT "[1] Oyun Kayıt İşlemi"
120 LOCATE 6, 25: PRINT "[2] Oyun Arama İşlemi"
140 LOCATE 7, 25: PRINT "[3] Oyun Silme İşlemi"
150 LOCATE 8, 25: PRINT "[4] Oyun Değiştirme İşlemi"
170 LOCATE 9, 25: PRINT "[5] Programdan Çıkış"
180 LOCATE 10, 25: PRINT "
190 LOCATE 11, 25: PRINT " Yazılım: Murat Saygı"
200 LOCATE 12, 25: PRINT " Ver. 1.0 1992 Tel: 9-343-33790"
210 LOCATE 13, 25: PRINT " Adres: Alişahane Mah. 196. Sok."
220 LOCATE 14, 25: PRINT " B. Blok No:2/4 70100 KARAMAN"
225 LOCATE 15, 25: PRINT "
227 LOCATE 16, 25: PRINT " Seçiminiz?..":
250 SS = INPUT$(1)
260 IF SS = "1" THEN GOTO 340
270 IF SS = "2" THEN GOTO 630
290 IF SS = "3" THEN GOTO 1060
300 IF SS = "4" THEN GOTO 1350
320 IF SS = "5" THEN GOTO 1860
330 GOTO 250
340 REM ..... KAYIT İŞLEMİ .....
350 CLS
360 PRINT SPC(30); "Oyun Kayıt İşlemi"
370 PRINT STRINGS(79, 196)
375 GOSUB 2000
380 LOCATE 3, 40: INPUT "Oyun Kodu.....": MK%: IF MK% = 0 THEN 70
390 GET #1, MK%
400 IF OZS = "" THEN LOCATE 4, 40: PRINT "Bu Kayıt Var !!!": GOTO 500
410 GOSUB 1900
470 LSET OAS = OS: LSET AKŞ = ACS: LSET AS = DS: LSET CS = FS: LSET OZS = ""
480 PUT #1, MK%
490 PRINT STRINGS(79, 196)
500 LOCATE 19, 40: PRINT " DEVAM (E/H):?"
590 SS = INKEY$: IF SS = "" THEN 590
600 IF SS = "E" OR SS = "e" THEN GOTO 340
610 IF SS = "H" OR SS = "h" THEN GOTO 70
620 GOTO 590
630 REM ..... ARAMA .....
640 CLS
650 PRINT SPC(30); "Oyun Arama İşlemi"
660 PRINT STRINGS(79, 196)
670 LOCATE 3, 1: INPUT "Aranışınızın Oyun Kodu:", MK%: IF MK% = 0 THEN 70
680 GET #1, MK%
690 IF OZS <> "" THEN PRINT "Bu Kayıt Yok !!": GOTO 760
700 GOSUB 2000
750 PRINT STRINGS(79, 196)
760 LOCATE 19, 40: PRINT " DEVAM (E/H):?"
850 SS = INKEY$: IF SS = "" THEN 850
860 IF SS = "E" OR SS = "e" THEN GOTO 630
870 IF SS = "H" OR SS = "h" THEN GOTO 70
880 GOTO 850
1060 REM ..... SİLME .....
1345 GOTO 70
1350 REM ..... DEĞİŞTİRME .....
1690 GOTO 70
1860 REM ..... OYUN KAYIT ÇIKIŞI .....
1870 CLS : LOCATE 10, 35: PRINT " BİTTİ "
1880 CLOSE : END
1900 REM ..... ALAN BİLGİLERİ OKU .....
1910 LOCATE 4, 40: INPUT "Oyun İsmi.....": O$
1920 LOCATE 5, 40: INPUT "Oyun Açıklaması.": ACS
1930 LOCATE 6, 40: INPUT "Oyun Uzunluğu.": DS
1950 LOCATE 7, 40: INPUT "Oyun Fiyatı.....": FS
1960 RETURN
2000 REM ..... ALAN BİLGİLERİ GÖSTER .....
2010 LOCATE 4, 1: PRINT "Oyun İsmi.....": OAS
2020 LOCATE 5, 1: PRINT "Oyun Açıklaması.": AKŞ
2030 LOCATE 6, 1: PRINT "Oyun Uzunluğu.": AS
2050 LOCATE 7, 1: PRINT "Oyun Fiyatı.....": CS
2060 RETURN
5000 RESUME 70
```